(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
27 November 2003 (27.11.2003)

PCT

(10) International Publication Number
WO 03/098479 A2

(51) International Patent Classification[7]: G06F 17/30

(21) International Application Number: PCT/US03/14892

(22) International Filing Date: 8 May 2003 (08.05.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/378,018 10 May 2002 (10.05.2002) US
10/254,383 24 September 2002 (24.09.2002) US

(71) Applicant: **ORACLE INTERNATIONAL CORPORATION** [US/US]; 500 Oracle Parkway, Redwood Shores, CA 94065 (US).

(72) Inventors: **YALAMANCHI, Aravind**; 8 Louisburg Square, Apt. 9, Nashua, NH 03060 (US). **GAWLICK, Dieter**; 757 Paul Avenue, Palo Alto, CA 94306 (US). **SRINIVASAN, Jagannathan**; 1 Hampshire Drive, Apt. F, Nashua, NH 03062 (US).

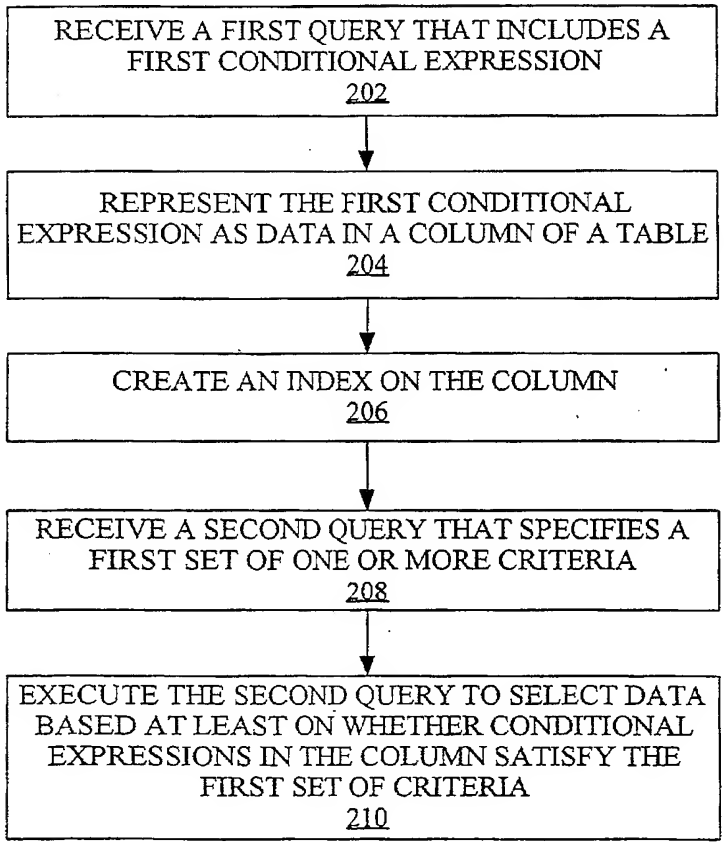
(74) Agents: **HENKHAUS, John** et al.; HICKMAN PALERMO TRUONG & BECKER LLP, 1600 Willow Street, San Jose, CA 95125 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE,

[Continued on next page]

(54) Title: MANAGING EXPRESSIONS IN A DATABASE SYSTEM



(57) Abstract: Managing expressions includes receiving a first query that includes a conditional expression. The expression is then represented as data in a column of a table. A second query is received that specifies a first set of criteria, and the second query is executed to select data based at least on whether expressions in the column satisfy the first set of criteria. In an embodiment, the second query further specifies a second set of criteria, wherein executing the second query includes selecting data based on whether data in columns other than the expression column satisfy the second criteria. A special index is defined, which can be created on the column that stores the expressions, to filter large sets of expressions efficiently. A method of evaluating an expression set stored as data in a table classifies each predicate from each expression, and filters the expression set based on the predicate classification.

WO 03/098479 A2

BNSDOCID: <WO_____03098479A2_I_>

ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

MANAGING EXPRESSIONS IN A DATABASE SYSTEM

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application is related to and claims the benefit of priority from U.S. Provisional Patent Application No. 60/378,018 entitled "Managing Expressions In A Database System," filed on May 10, 2002, which is incorporated by reference in its entirety for all purposes, as if fully set forth herein.

FIELD OF THE INVENTION

[0002] The present invention relates generally to database management systems and, more specifically, to techniques for managing conditional expressions in database systems.

BACKGROUND OF THE INVENTION

[0003] In the context of event and content-based subscription systems, events are defined which, when met, trigger an action. For example, a subscriber can define rules that include events that define a state of content that, when met, trigger transmission of content to the subscriber. Using a database management system as an underlying engine for an event-based subscription system, a subscriber can register queries with the system that represent conditional expressions on the content of the events. In such a subscription or similarly functioning system, a potentially very large set of queries, i.e., an expression set on the content, are registered to manage the publication of desired content data. When a given data item becomes available, these conditional expressions are filtered to find those expressions that match the given data item.

[0004] A simple but inefficient approach to the task of filtering expression sets is to test all of the expressions in a given set for each data item. However, this approach is scalable neither for a large set of expressions nor for a high rate of events. Therefore, most commercial systems pre-process the expression set and create in-memory matching networks (i.e., specialized data structures) that group matching predicates in the expression set and share the processing cost across multiple expressions.

[0005] Matching networks are decision trees in which each node represents a predicate group in a given expression set. Data flows from a parent node to its children only if the data evaluates to true for the predicate representing the parent node. A path from the root of the decision tree to a leaf node represents all the conjunctions in an expression. The leaf nodes in the tree are labeled with expression identifiers and if a data

BNSDOCID: <WO_____03098479A2_1_>

item passes the predicate test on a leaf node, the corresponding expressions are considered true for that data item. Many variants of the matching networks (like RETE, TREAT and Gator networks) are in use for the existing systems.

[0006] In existing systems, any operation requiring filtering of expressions and related information requires significant custom coding and reduces performance characteristics. Furthermore, the number of expressions is limited in size as the corresponding matching networks must fit in main-memory, changes in expressions are costly, and users are unable to adjust filtering strategies to the structure and use of the expressions and related data.

[0007] Based on the foregoing, it is clearly desirable to provide an improved mechanism for managing expressions, such as expressions associated with a subscription system. In addition, there is a more specific need for a mechanism that provides the ability to filter expressions in conjunction with filters on other related information.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] The present invention is illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings and in which like reference numerals refer to similar elements and in which:

[0009] FIG. 1 is an example table used to support examples of processes or steps described herein;

[0010] FIG. 2 is a flowchart that illustrates a process for managing expressions in a database;

[0011] FIG. 3A is a flowchart that illustrates a process for evaluating an expression set that is stored as data in a column of a table;

[0012] FIG. 3B is a flowchart that illustrates a step of filtering expressions based on predicate classifications; and

[0013] FIG. 4 is a block diagram that illustrates a computer system upon which an embodiment of the invention may be implemented.

DETAILED DESCRIPTION

[0014] A method and system are described for managing expressions in a database system. In addition, and more specifically, methods and systems are described for managing conditional expressions associated with event and content-based information subscription systems.

BNSDOCID: <WO_____03098479A2_I_>

[0015] In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring the present invention.

OVERVIEW

[0016] Conditional expressions, often in the form of standard database query, are represented as data in a column of a table. These expressions may represent, for example, data filters for filtering data in an information subscription system. Another standard database query, which specifies criteria, can then be executed on the column to determine whether expressions in the column meet the specified criteria. The criteria may represent, for example, incoming data to a subscription system.

[0017] Thus, the expression processing mechanism described herein is integrated with database technology by treating the expressions as data that can be queried along with other related user-specified data. For example, a publisher may include in the query other filtering criteria related to its subscribers.

[0018] In an embodiment, a first query is received that includes a first conditional expression. For example, the first query may be received via INSERT or database load operations. The first expression is then represented as data in a column of a table. A second query is received that specifies a first set of criteria, and the second query is executed to select data based at least on whether expressions in the column satisfy the first set of criteria. For example, the second query may be received from a user of a database application.

[0019] In an embodiment, the second query further specifies a second set of criteria, wherein executing the second query includes selecting data based on whether data in columns other than the column satisfy the second criteria.

[0020] Other embodiments include receiving the first query from a subscriber to an subscription service or system, wherein the first expression specifies criteria that must be satisfied by data for the data to be of interest to the subscriber; and receiving the second query from a publisher associated with the subscription system, wherein the second query specifies criteria that must be satisfied by data, with respect to the subscriber, for the data to be published to the subscriber. Thus, mutual filtering can be performed on data by both subscribers and publishers.

BNSDOCID: <WO_____03098479A2_I_>

[0021] According to one aspect, a special index is defined on the column that stores the expressions as data, to filter large sets of expressions efficiently.

[0022] In an embodiment, predicates in the expressions are grouped based on the commonality of their left-hand sides, i.e., common identifiers associated with criteria of the first expressions. These groups are persistently stored in a database. Further, operators and constants associated with the predicates are stored in a predicate table in association with respective predicate groups, and an index, such as a concatenated bitmap index, is created based on the operators and constants.

MANAGING EXPRESSIONS IN A DATABASE SYSTEM

[0023] Typically, in event and content-based subscription systems, a set of expressions or rules (e.g., Event-Condition-Action (ECA) rules) is defined for an event structure or context, and the structure of the event determines the elementary attributes that can be used in the expressions. The term "event" is used in this context to refer to the data item for which expressions, or "conditions", are evaluated. For example, an event, in the context of stock trading, might be a publicly-traded stock attaining a particular price, which is represented as a data item. Thus, elementary attributes of expressions associated with such an event could include, for example, SYMBOL, PRICE, and CHANGE. An ECA rule lies dormant until it is triggered by the occurrence of an Event.

[0024] A set of expressions defined for an event structure or context is called an expression set. For example, Subscriber A may be interested in an event expressed as SYMBOL = ORCL, PRICE > 30, and CHANGE > 5; Subscriber B may be interested in an event expressed as SYMBOL = CSCO, and PRICE > 25; and Subscriber C may be interested in an event expressed as SYMBOL = INTC, and CHANGE/PRICE > 0.1.

[0025] In a typical system based on rules, for example, a content-based subscription system, efficient filtering of a large set of conditional expressions is critical for the scalability of the system. Unlike a typical database design, where a few queries are executed on a large set of rows in a table, a rules-based system has a large number of expressions (similar to WHERE clause of a query) operating against a single data item.

[0026] The expressions defined for an application are relatively static and the rate at which the new data should be processed against these expressions is high. Therefore, according to one aspect, pre-processing of the expressions is performed to build specialized data structures and data access mechanisms such as indexes, to allow fast filtering with a new data item.

BNSDOCID: <WO_____03098479A2_I_>

PRE-PROCESSING A SET OF EXPRESSIONS

[0027] Given a large number of conditional expressions in a system, the expressions tend to have similarities at the elementary predicates level. That is, two expressions, each with one or more elementary predicates joined by conjunctions, may have a common predicate. Thus, expression evaluation costs are shared among multiple expressions, thereby leveraging the foregoing similarities. In an embodiment, a library of functions, the Expression Type Library, supports the pre-processing of a set of expressions.

[0028] The Expression Type Library provides the basic functionality required for building a matching network for expressions. This library is supplied as a set of Java classes which are capable of converting an expression string into an expression tree containing elementary predicates, joined by conjunctions and disjunctions. This library can be used by any indexing scheme for pre-processing the expressions in an expression set and during incremental modifications to these expressions.

[0029] This library parses the expressions and processes them as follows :

[0030] (1) Normalize the expression, for example, by rearranging the predicates in an expression to rewrite it in a disjunctive normal form (DNF), that is, an OR list of AND sub-clauses. For example, an expression of form

SYMBOL = 'GE' and (PRICE < 25 or PRICE > 35)

is rewritten as follows after a DNF conversion:

(SYMBOL = 'GE' and PRICE < 25) or (SYMBOL = 'GE' and PRICE > 35).

[0031] (2) Normalize the predicates, by rewriting each predicate in an expression such that it has a pure constant on the right-hand side. For example, a predicate of form

PRICE > 27 + CHANGE

is rewritten as:

PRICE - CHANGE > 27

[0032] (3) Decode the predicate, by resolving it as follows:

left-hand side (LHS): an expression or a function of one or more attributes (e.g., PRICE - CHANGE from the predicate PRICE - CHANGE > 27);

right-hand side (RHS): a constant (e.g., the "27" from the predicate PRICE - CHANGE > 27); and

relational operator: the relational operator joining the LHS and RHS (e.g., the ">"from the predicate PRICE - CHANGE > 27).

[0033] The left-hand side of a predicate, for example, the attribute SYMBOL, is also referred to as a complex attribute. It could also be a sub-expression involving one or more elementary attributes or user-defined functions, for example, CHANGE/PRICE.

BNSDOCID: <WO_____03098479A2_I_>

The operators of a predicate are mapped to a predetermined integer value, however, embodiments are not limited to such a mapping. The mapped operators and associated predicate constants (RHS) are stored in a predicate table, which is described in more detail below. In an implementation, predicates involving constructs such as IN lists, sub-queries, etc. are not grouped with other predicates.

ATTRIBUTE SET

[0034] According to an aspect of the invention, an attribute set that captures the event structure or context is created. The attribute set has a list of elementary attributes used in an expression set, and their data types. In its simplest form, the attribute set resembles an abstract type definition in the database. For example, a set of elementary attributes used in an expression set constitutes its attribute set.

[0035] One non-limiting technique for creating an attribute set is expressed in the following commands:

```
EXECUTE dbms_expeng.create_attribute_set (
                    attr_set => 'TICK');
EXECUTE dbms_expeng.add_elementary_attribute (
                    attr_set  => 'TICK',
                    attr_name => 'SYMBOL',
                    attr_type => 'VARCHAR2(6)');
EXECUTE dbms_expeng.add_elementary_attribute (
                    attr_set  => 'TICK',
                    attr_name => 'PRICE',
                    attr_type => 'NUMBER');
EXECUTE dbms_expeng.add_elementary_attribute (
                    attr_set  => 'TICK',
                    attr_name => 'CHANGE',
                    attr_type => 'NUMBER');
```

whereby an attribute set "TICK" is created, having elementary attributes "SYMBOL", "PRICE", and "CHANGE".

[0036] Significantly, the expression set is stored in a column of a table. For example, expressions can be stored in a VARCHAR2 or a CLOB column in a database table. Such a column can be recognized as a column of EXPRESSION data type by, for example, associating some expression set metadata to the column. Furthermore, the column storing

BNSDOCID: <WO_____03098479A2_I_>

the expression set is associated with the attribute set created for the expression set. One non-limiting technique for associating an attribute set with an expression set is expressed in the following commands:

```
EXECUTE dbms_expeng.assign_attribute_set (
                attr_set   => 'TICK',
                tab_name   => 'TRADER',
                exp_column => 'INTEREST');
```

whereby the attribute set "TICK" is associated with an expression set stored in column "INTEREST" of table "TRADER".

[0037] A VARCHAR2 or a CLOB column associated with an attribute set constitutes an EXPRESSION column. The values stored in an EXPRESSION column are treated as expressions and they are initially expected to adhere to SQL-WHERE clause format that can include XPATH expressions. These expressions can use all the attributes defined in the attribute set along with any system variables and user-defined functions that are valid in the user environment, for example,

```
UPPER (symbol) = 'INTC' AND change/price > 0.1.
```

EXPRESSION FILTER

[0038] According to an embodiment, an Expression Filter is a set of PL/SQL packages and APIs to manage expressions in user tables, and to filter the expressions for a given data item, that is, to match criteria expressed in expressions with the given data item, using a standard SQL or other query language query. In a publication system, the expressions specify criteria that must be satisfied by data for the data to be of interest to a subscriber. The Expression Filter comprises two components: an EVALUATE operator (described immediately below) and an Expression Filter Indextype (described under the heading "Creating An Index For The Expression Set").

EVALUATE OPERATOR

[0039] A new operator is introduced that processes the expression set stored in an EXPRESSION column. This operator can be used in the WHERE clause of a standard SQL, or a DML statement, to filter the expressions for a data item. The predicate on the expression set, using the new operator EVALUATE, can be combined with any other predicate on the table. The EVALUATE operator accepts the name of the column storing the expressions, e.g., INTEREST, and a given data item as arguments. The EVALUATE operator internally uses the expression set metadata to evaluate expressions for data items passed in.

BNSDOCID: <WO_____03098479A2_I_>

[0040] An example of a query that uses the EVALUATE operator is as follows:

```
SELECT * FROM traders
WHERE EVALUATE (traders.exp,
    'symbol=>"intc", price=>32, change=>3.3') = 1
AND traders.city = 'New York'.
```

The expression is considered true if the query evaluates to true for given data item values.

[0041] The query on the table in which expressions are stored can be extended to include multi-table joins and any other database query operations using GROUP BY clause, ORDER BY clause, HAVING clause etc. In addition, filtering a set of expressions for a batch of data items by joining the table in which expressions are stored with the table storing the data items is contemplated. An example of such a query is as follows:

```
SELECT distinct (dataitems.symbol), count(*) FROM traders, dataitems
WHERE EVALUATE (traders.exp,
            tick.getVarchar(dataitems.symbol,
                        dataitems.price,
                        dataitems.change)) = 1
AND traders.city = 'New York'
GROUP BY dataitems.symbol;
```

where TICK is the name of the attribute set defined for the expression set, TRADERS is a table that stores the expression set (see FIG. 1 for an example), and DATAITEMS is a table that stores the data items being processed, i.e., being compared to the expression set.

CREATING AN INDEX FOR THE EXPRESSION SET

[0042] Testing every expression for a data item is a linear time solution. When a large set of expressions are defined, this approach is not scalable for a high volume of data items. Therefore, in an embodiment, a new indexing mechanism is used to evaluate a large set of expressions efficiently and, consequently, to quicken the evaluation of the expression set for a given data item or data string. This index can be defined on an EXPRESSION column, thus a query optimizer can determine the use of the index for the evaluation of an expression set, based on the index usage cost. In an implementation, persistent database objects are created to maintain the index for an expression set, where pre-processing the expressions set at the time of index creation populates these database objects. Additionally, the information stored in these objects is maintained to reflect any changes to the expression set using DML operations on the table storing the expressions.

BNSDOCID: <WO_____03098479A2_I_>

EXPRESSION FILTER INDEX TYPE

[0043] In an embodiment, the indexing scheme is implemented as a new Indextype, Expression Filter, using an extensible indexing framework. In an implementation, the indexing scheme is implemented as a new Indextype, Expression Filter, using the Oracle Extensible Indexing framework. The Expression Filter index type can be used to create an index on any set of expressions stored in a database column of type VARCHAR2, CLOB or BFILE. However, use of another index type other than the foregoing, which may be used on expressions stored as data types, is contemplated and therefore within the scope of embodiments of the invention.

[0044] The Expression Filter index type implementation parses a set of expressions and groups the predicates in the expressions into disjoint sets with matching Left-Hand-Sides. The data structures used to group the predicates in an expression set are relational in nature. In an implementation in which persistent database objects are used for the Expression Filter index data structure, an example of such database objects are as follows:

Predicate Table: a relational table that stores the predicates appearing in the expressions;

Bitmap Indexes: one or more bitmap indexes on the predicate table; and

Access Function: a function that queries the predicate table to filter the expressions for a data item.

[0045] These objects collectively provide for efficient filtering of expressions based on both equality and range predicates. Furthermore, since the index structure objects are persistently stored in the database, memory constraints associated with the size of expression sets in prior approaches, which typically use main memory extensively, are not applicable to the present embodiments. By contrast, operations using the present embodiments can store the necessary database blocks into a database buffer cache as they are needed.

[0046] As described above, expressions from a given set of expressions refer to a set of elementary attributes with fixed data types. Further, a set of valid values for these attributes constitute a data item, which is evaluated against these expressions. Hence, to index a set of expressions with the Expression Filter index type, all the elementary attributes used in the expression set should be associated with the database column storing the expressions. These elementary attributes, along with some optional complex attributes, constitute the attribute set for the expression set, which are stored in one or more data dictionary tables. One non-limiting technique for creating an Expression Filter

BNSDOCID: <WO_____03098479A2_I_>

index as described above is expressed in the following commands, which create an index TRADERFILTER on the INTEREST column of table TRADER in which the expressions are stored. Additionally, the index is configured to filter the predicates involving SYMBOL and PRICE attributes efficiently by defining bitmap indexes on such attributes.

```
CREATE INDEX traderfilter ON trader ( interest )
    INDEXTYPE IS EXPFIL.EXPFILTER
    PARAMETERS ('STOREATTRS (symbol, price, change)
        INDEXATTRS (symbol, price)').
```

[0047] The parameters passed to the CREATE INDEX statement determine the structure of the objects (predicate table, bitmap indexes and access function) used for the Expression Filter index and, therefore, influence the performance of the filter. The PARAMETERS clause in the CREATE INDEX statement is optional and in the absence of this clause, all the elementary attributes (with native types) in the attribute set are stored and indexed.

[0048] In an embodiment, a concatenated bitmap index is created on the predicate table's operator and RHS constant columns for frequent LHSs. For example, the LHS attribute SYMBOL is frequently used in expressions regarding stock quotes, so a bitmap index might be created on operator and constant columns associated with predicates that include the SYMBOL attribute, and combined into a concatenated bitmap index. These bitmap indexes function as a multi-dimensional index during the expression set evaluation.

[0049] Generally, utilization of the indexing scheme described herein, implemented as a new Indextype (i.e., Expression Filter index type) and applied to a column storing expressions as data, provides a method for efficiently evaluating large sets of expressions by allowing the new EVALUATE operator to use the Expression Filter index.

PREDICATE EVALUATION

[0050] In an embodiment, in analysis of predicates, the predicates in an expression set are classified into three sets:

(1) Predicates with indexed attributes: bitmap indexes are created for the predicate groups belonging to this set, for example, popular predicate such as predicates that include the SYMBOL identifier;

(2) Predicates with Stored attributes: the predicate groups belonging to this set are parsed and stored in the predicate table with no indexes defined on the <operator, RHS constant> columns; and

BNSDOCID: <WO_____03098479A2_I_>

(3) Sparse predicates: the predicates belonging to this set are stored in their original form. If more than one sparse predicate exists for an expression, they are combined into one conjunction. Note that it is not always efficient to add a new set of columns to the predicate table for every predicate that cannot be grouped with others. Hence, according to an embodiment, a separate VARCHAR2 column, PARTIAL_CONDITION, is defined in the predicate table to hold the conditional expression for sparse predicates. Unlike other columns in the predicate table, this column is not indexed and it can hold one or more predicate definitions in conjunctive form.

[0051] A predicate falls into one of the above sets based on the cost of computing its left-hand side and the frequency of occurrence of its left-hand side in the expression set. The evaluation cost for a predicate depends on the set it belongs to.

[0052] Steps involved in evaluating a predicate, with respect to its classification as described above, are as follows.

(1) Indexed attribute: a one-time computation of the complex attribute (i.e., LHS of the predicate group), and one or more range scans on the bitmap indexes using the computed value;

(2) Stored attribute: a one-time computation of the complex attribute, and comparison of the computed value with the RHS of all the predicates in this group; and

(3) Sparse predicate: parsing of the sub-expression representing the sparse predicate, and evaluation of the sub-expression through substitution of data values.

[0053] During the expression set evaluation, according to an embodiment, the expressions are filtered in three phases, as follows.

[0054] Phase 1: The predicates belonging to the Indexed attribute set are tested by performing a few range scans on the bitmap indexes defined thereon. The results from these scans are combined (bitmap AND operation) to obtain a set of expressions for which these predicates are all true for the given data item.

[0055] The total cost of this phase is defined as

$I * C11 + N1 * C12 * \log(E1) + C13 * E2;$

where

I is the number of bitmap indexes used for filtering;

E1 is the number of expressions to be filtered;

C11 is the average bitmap index processing cost per index;

C12 * log(E1) is the average cost of one index lookup;

C13 is the cost of fetching one row from the predicate table using the row identifier;

BNSDOCID: <WO_____03098479A2_I_>

E2 is the number of expressions evaluated to true in Phase 1; and

N1 is the average number of lookups per bitmap index.

[0056] The result of Phase 1 is a set of expressions that evaluate to true with all the predicates belonging to the indexed attribute set.

[0057] Phase 2: For all the expressions evaluated to true in Phase 1, the predicates belonging to the Stored attribute set are tested.

[0058] The total cost of this phase is defined as

S * C2 * E2;

where

S is the average number of stored attributes per expression;

C2 is the average cost of one comparison;

E2 is the total number of expressions evaluated to true in Phase 1. Typically, the working set is narrowed down considerably in Phase 1 and E2 << E1.

[0059] The result of Phase 2 is a set of expressions that evaluate to true with the predicates belonging to the indexed or stored attribute sets.

[0060] Phase 3: For all the expressions that are true after Phase 2, the sparse predicates (if any) for these expressions are evaluated.

[0061] The total cost of this phase is defined as follows

P * C3 * E3;

where

P is the probability of a sparse predicate for an expression;

C3 is the average cost of parsing and evaluating a sparse predicate; and

E3 is the total number of expressions that are true after Phase 2 of filtering.

[0062] The result of Phase 3 is a set of expressions that evaluate to true for the given data item. Note that alternative access plans other than the index-based evaluation described above can be used.

[0063] Consider a database table TRADERS as illustrated in FIG. 1, which stores as conditional expressions information about stock traders and their interest in trading. In order to find all the TRADERS who are under 30 years of age, living in New York, and interested in a data item (SYMBOL='ORCL', PRICE=31, CHANGE=5.2), the following query is an example of a query that can be issued on the TRADERS table.

```
SELECT name, phone FROM traders
    WHERE EVALUATE (interest,
        'SYMBOL='ORCL', PRICE=>31, CHANGE=>5.2') = 1 AND
    city = 'New York' AND age < 30;
```


BNSDOCID: <WO_____03098479A2_I_>

[0064] In practice, this query could make use of the predicate table and the bitmap indexes on the predicate table to filter the expressions. The Expression Filter index returns the rowids for all the expressions that evaluate to true with the given data item. Remaining predicates in the SELECT statement are evaluated on the corresponding rows to answer the query. In addition to the Expression Filter index on the INTEREST column, if the TRADERS table has native or extensible indexes defined on the CITY and AGE columns, an optimizer program may choose one or more of these indexes, based on the access cost, to answer the above query.

[0065] When a new row is inserted into the TRADERS table or some of the existing expressions are updated, the Expression Filter index is automatically maintained to reflect these changes.

[0066] As the foregoing example query illustrates, the techniques described herein integrate expression filtering operations into database operations. In the context of an information subscription-publication system, as a result of the integration of expressions and filtering mechanisms into a database system, mutual filtering from both a subscriber and a publisher can be performed, and performed efficiently, through use of standard SQL statements. Therefore, multi-domain queries are possible, by joining tables and adding predicates to a query to further filter expressions. Furthermore, batch evaluation of expression sets for a given set of data is possible by joining the table storing the expressions with the table storing the data items.

[0067] Prior approaches separate characteristics of system users or clients, such as name, telephone number and residence, from their respective interests in data, which is expressed in subscription expressions. Many applications can benefit from this integration of interests and personal characteristics. For example, a query could be executed to return "all traders in New York with an interest in Stock X exceeding a price of Y" or "all customers located within 10 miles of my store with an interest in vehicle Z." Significantly, present embodiments facilitate optimal filtering of expressions based on their context. For example, in the context of stock trading, this context leads to efficient filtering of data by grouping predicates on the SYMBOL attribute, since the vast majority of users will have interests relative to a specific stock represented by a stock symbol, and by indexing operators and constants associated with predicates that include the SYMBOL attribute.

BNSDOCID: <WO_____03098479A2_I_>

PROCESS FOR MANAGING EXPRESSIONS IN A DATABASE

[0068] FIG. 2 is a flowchart that illustrates a process for managing expressions in a database. FIG. 2 is described with additional reference to the table of FIG. 1.

[0069] At block 202, a first query is received that includes a first conditional expression. For example, the first query may be received via INSERT or database load operations. In an embodiment, conditional expressions are expected to be in the form of a SQL WHERE clause. Further, any SQL WHERE clause can be treated as an expression. In alternative embodiments, conditional expressions having a form other than a SQL query, and having a form other than any type of query, are contemplated. For example, a user may specify interests in data via simple textual input, which is converted directly into an appropriate data format, such as VARCHAR2 or CLOB, for storing in a column of a table.

[0070] In an embodiment, the first query is received from a subscriber to an information subscription system, or other event or content-based information publication system, wherein the expression specifies criteria that must be met by data for the data to be of interest to a particular subscriber. Thus, the first query can serve as a mechanism for a first level of filtering with respect to system data, essentially processing data for dispatch or publication to appropriate subscribers.

[0071] At block 204, the first conditional expression of the first query is expressed as data in a column of a table, as described above and as depicted as column 106 of FIG. 1. For example, one or more predicates constituent to the expression are encoded in a data format on which SQL or another query language can operate, and on which a database management system can manage. Note that the representation of expressions in the INTEREST column 106 of the table of FIG. 1 is for purposes of explanation, and do not take the form as depicted. Rather, the expressions stored in the INTEREST column, or any similar column in which expression representations are stored, are actually conventionally encoded into an appropriate data format.

[0072] At optional block 206, an index is created on the column of the table in which the conditional expressions are stored as data at block 204. In this embodiment, an index is created as described above.

[0073] Indexing steps include grouping predicates from a set of expressions, based on predicate identifiers that are associated with respective criteria of the first set of criteria. Continuing with the example, predicate group identifiers might include SYMBOL, PRICE, and CHANGE. Sets of predicate operators and constants are stored in a predicate table in association with respective predicate identifiers. For example, a predicate

BNSDOCID: <WO_____03098479A2_I_>

grouping for SYMBOL may have entries in the predicate table representing sets of operator and constant combinations, such as operator is "equal to" and constant is "ORCL", and a predicate grouping for PRICE may have entries in the predicate table that include operator and constant combinations such as "greater than or equal to" and "31". These sets of operators and constants can be encoded in any conventional manner appropriate for their purpose. Next, a concatenated bitmap index, or another form of index, is created based on a set of predicate operators and constants associated with a particular predicate identifier. For example, a set of operator and constant combinations, represented in respective columns of a predicate table, for the predicate identifier SYMBOL, is indexed for fast and efficient evaluation of predicates and thus, expressions, as part of a data filtering process.

[0074] At block 208, a second query is received that specifies a first set of one or more criteria. The following query is an example of a second query.

```
SELECT name, phone FROM traders
    WHERE EVALUATE (interest,
            'SYMBOL='ORCL', PRICE=>31, CHANGE=>5.2) = 1 AND
        city = 'New York' AND age < 30;
```

This query specifies the criteria, SYMBOL is equal to "ORCL", PRICE is greater than or equal to 31, and the CHANGE in PRICE is greater than or equal to 5.2. The second query may be received, for example, from a user of a database application.

[0075] In an embodiment, the second query is received from a publisher in an information subscription system, or other event or content-based information publication system, wherein the expression specifies a set of criteria that must be met by data for the data to be published to a particular subscriber. Thus, the second query can serve as a mechanism for a second level of filtering with respect to system data, essentially processing data for dispatch or publication to appropriate subscribers.

[0076] At block 210, the second query is executed to select data based at least on whether conditional expressions in column 106 satisfy the first set of criteria. For example, execution of the second query determines, among other things, whether a particular data item meets a condition or set of conditions as expressed in a stored expression in the data column 106. In other words, in the context of a subscription system, it is determined whether there are any subscribers that are interested, through the conditions or predicates specified in their expressions, in data meeting the criteria specified in the first part of the WHERE clause (before the AND conjunction) of the query.

BNSDOCID: <WO_____03098479A2_I_>

[0077] In an embodiment, the second query further specifies a second set of criteria, wherein execution of the second query at block 210, includes selecting data based on whether data in one or more columns other than the expression column, such as columns 102, 104, satisfies the second criteria. For example, the second part of the WHERE clause (after the AND conjunction) specifies that other columns, CITY and AGE, are to be considered to determine whether a given data item being processed additionally satisfies the criteria of CITY (column 104) equals "New York" and AGE (column 102) is less than 30. The index created at block 206 is optional, but in instances in which an index is created, execution of the second query can utilize the index to evaluate the conditional expressions efficiently.

[0078] In practice, the second query can operate as a mutual filtering mechanism for considering more than one level or direction of filtering. For example, the second query above considers both subscribers' and publishers' data filtering interests. Significantly, the process described can operate in an information security related implementation, with the second set of criteria expressing security or authorization criteria regarding to whom particular information can or should be published or provided.

PROCESS FOR EVALUATING AN EXPRESSION STORED AS DATA IN A COLUMN OF A DATABASE TABLE

[0079] FIG. 3A is a flowchart that illustrates a process for evaluating an expression set that is stored as data in a column of a table.

[0080] At block 302, each predicate form each expression of the expression set is classified as one of an indexed attribute predicate, a stored attribute predicate, and a sparse predicate. An indexed attribute predicate is a predicate that is chosen to be indexed, as described above, wherein the index is based on a set of operator and constant attributes that are stored in respective columns of a predicate table in association with a respective predicate identifier.

[0081] A stored attribute predicate is a predicate for which its operator and constant attributes are stored in respective columns of the predicate table in association with a respective identifier, but for which no index is created. For example, stored attribute predicates might not be indexed because their identifiers are not commonly used enough in the expression set, or they might be complex predicates that comprise operations with basic attributes (e.g., CHANGE/PRICE).

[0082] A sparse predicate, as described above, is a predicate in which a predicate identifier associated therewith is uncommon in the expression set. For example, a sparse

BNSDOCID: <WO_____03098479A2_I_>

predicate might be derived from a predicate with a BETWEEN operator, with multiple operators on the same attribute, with functions, and with a LIKE operator. A sparse predicate, and its associated operator and constant, is stored as data in the predicate table.

[0083] At block 304, the expression set is filtered based on the classification of predicates performed at block 302. FIG. 3B is a flowchart that illustrates the step of filtering expressions based on predicate classifications, block 304.

[0084] At block 304A, the indexed attribute predicates are filtered first to obtain a first set of expressions that includes one or more expressions for which all of its indexed attribute predicates are true for a given data item. Second, at block 304B, the stored attribute predicates of the first set of expressions are filtered to obtain a second set of expressions that include one or more expressions for which all of it stored attributes are true for the given data item. Third, at block 304C, the sparse predicates of the second set of expressions, if any, are filtered to obtain a third set of expressions that includes one or more expressions for which all of its predicates are true for the given data item. Hence, the data item meets the criteria specified in the expression.

HARDWARE OVERVIEW

[0085] FIG. 4 is a block diagram that illustrates a computer system 400 upon which an embodiment of the invention may be implemented. Computer system 400 includes a bus 402 or other communication mechanism for communicating information, and a processor 404 coupled with bus 402 for processing information. Computer system 400 also includes a main memory 406, such as a random access memory (RAM) or other dynamic storage device, coupled to bus 402 for storing information and instructions to be executed by processor 404. Main memory 406 also may be used for storing temporary variables or other intermediate information during execution of instructions to be executed by processor 404. Computer system 400 further includes a read only memory (ROM) 408 or other static storage device coupled to bus 402 for storing static information and instructions for processor 404. A storage device 410, such as a magnetic disk, optical disk, or magneto-optical disk, is provided and coupled to bus 402 for storing information and instructions.

[0086] Computer system 400 may be coupled via bus 402 to a display 412, such as a cathode ray tube (CRT) or a liquid crystal display (LCD), for displaying information to a computer user. An input device 414, including alphanumeric and other keys, is coupled to bus 402 for communicating information and command selections to processor 404. Another type of user input device is cursor control 416, such as a mouse, a trackball, or

BNSDOCID: <WO_____03098479A2_I_>

cursor direction keys for communicating direction information and command selections to processor 404 and for controlling cursor movement on display 412. This input device typically has two degrees of freedom in two axes, a first axis (e.g., x) and a second axis (e.g., y), that allows the device to specify positions in a plane.

[0087] The invention is related to the use of computer system 400 for implementing the techniques described herein. According to one embodiment of the invention, those techniques are performed by computer system 400 in response to processor 404 executing one or more sequences of one or more instructions contained in main memory 406. Such instructions may be read into main memory 406 from another computer-readable medium, such as storage device 410. Execution of the sequences of instructions contained in main memory 406 causes processor 404 to perform the process steps described herein. In alternative embodiments, hard-wired circuitry may be used in place of or in combination with software instructions to implement the invention. Thus, embodiments of the invention are not limited to any specific combination of hardware circuitry and software.

[0088] The term "computer-readable medium" as used herein refers to any medium that participates in providing instructions to processor 404 for execution. Such a medium may take many forms, including but not limited to, non-volatile media, volatile media, and transmission media. Non-volatile media includes, for example, optical, magnetic, or magneto-optical disks, such as storage device 410. Volatile media includes dynamic memory, such as main memory 406. Transmission media includes coaxial cables, copper wire and fiber optics, including the wires that comprise bus 402. Transmission media can also take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0089] Common forms of computer-readable media include, for example, a floppy disk, a flexible disk, hard disk, magnetic tape, or any other magnetic medium, a CD-ROM, any other optical medium, punchcards, papertape, any other physical medium with patterns of holes, a RAM, a PROM, and EPROM, a FLASH-EPROM, any other memory chip or cartridge, a carrier wave as described hereinafter, or any other medium from which a computer can read.

[0090] Various forms of computer readable media may be involved in carrying one or more sequences of one or more instructions to processor 404 for execution. For example, the instructions may initially be carried on a magnetic disk of a remote computer. The remote computer can load the instructions into its dynamic memory and send the instructions over a telephone line using a modem. A modem local to computer system

BNSDOCID: <WO_____03098479A2_I_>

400 can receive the data on the telephone line and use an infra-red transmitter to convert the data to an infra-red signal. An infra-red detector can receive the data carried in the infra-red signal and appropriate circuitry can place the data on bus 402. Bus 402 carries the data to main memory 406, from which processor 404 retrieves and executes the instructions. The instructions received by main memory 406 may optionally be stored on storage device 410 either before or after execution by processor 404.

[0091] Computer system 400 also includes a communication interface 418 coupled to bus 402. Communication interface 418 provides a two-way data communication coupling to a network link 420 that is connected to a local network 422. For example, communication interface 418 may be an integrated services digital network (ISDN) card or a modem to provide a data communication connection to a corresponding type of telephone line. As another example, communication interface 418 may be a local area network (LAN) card to provide a data communication connection to a compatible LAN. Wireless links may also be implemented. In any such implementation, communication interface 418 sends and receives electrical, electromagnetic or optical signals that carry digital data streams representing various types of information.

[0092] Network link 420 typically provides data communication through one or more networks to other data devices. For example, network link 420 may provide a connection through local network 422 to a host computer 424 or to data equipment operated by an Internet Service Provider (ISP) 426. ISP 426 in turn provides data communication services through the world wide packet data communication network now commonly referred to as the "Internet" 428. Local network 422 and Internet 428 both use electrical, electromagnetic or optical signals that carry digital data streams. The signals through the various networks and the signals on network link 420 and through communication interface 418, which carry the digital data to and from computer system 400, are exemplary forms of carrier waves transporting the information.

[0093] Computer system 400 can send messages and receive data, including program code, through the network(s), network link 420 and communication interface 418. In the Internet example, a server 430 might transmit a requested code for an application program through Internet 428, ISP 426, local network 422 and communication interface 418.

[0094] The received code may be executed by processor 404 as it is received, and/or stored in storage device 410, or other non-volatile storage for later execution. In this manner, computer system 400 may obtain application code in the form of a carrier wave.

BNSDOCID: <WO_____03098479A2_I_>

EXTENSIONS AND ALTERNATIVES

[0095] Alternative embodiments of the invention are described throughout the foregoing description, and in locations that best facilitate understanding the context of the embodiments. Furthermore, the invention has been described with reference to specific embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention. For example, implementations were presented in which SQL is used; however, the techniques described herein are not limited to use with SQL, for other data query languages may be applicable. For another example, implementations were presented in the context of a subscriber/publisher system; however, advantages and use of embodiments of the invention are not limited to this context. For one more example, implementations were presented in which a concatenated bitmap index is created on columns of a predicate table. However, embodiments are not limited to such an index, for other index types that are suitable for indexing multiple columns of data tables are also applicable. Therefore, the specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

[0096] In addition, in this description certain process steps are set forth in a particular order, and alphabetic and alphanumeric labels may be used to identify certain steps. Unless specifically stated in the description, embodiments of the invention are not necessarily limited to any particular order of carrying out such steps. In particular, the labels are used merely for convenient identification of steps, and are not intended to specify or require a particular order of carrying out such steps.

BNSDOCID: <WO_____03098479A2_I_>

CLAIMS

What is claimed is:

1. A method for managing expressions in a database, the method comprising the steps of:
receiving a first query that includes a first conditional expression;
representing the first conditional expression as data in a column of a table;
receiving a second query that specifies a first set of one or more criteria; and
executing the second query to select data based at least on whether conditional expressions in the column satisfy the first set of criteria.

2. The method of Claim 1, wherein the second query further specifies a second set of one or more criteria, and wherein the step of executing the second query includes selecting data based on whether data in one or more columns other than the column satisfies the second set of criteria.

3. The method of Claim 1, wherein the step of receiving a first query includes receiving the first query from a first user of a database application, and wherein the first conditional expression specifies one or more criteria that must be satisfied by data for the data to be of interest to the user.

4. The method of Claim 3, wherein the step of receiving the second query includes receiving the second query from a second user of a database application, and wherein the second query specifies a second set of one or more criteria that must be satisfied by data for the data to be made available to the first user, thus enabling mutual filtering among the first user and the second user.

5. The method of Claim 1, wherein the step of receiving a first query includes receiving the first query from a subscriber to an information subscription system, and wherein the first conditional expression specifies one or more criteria that must be satisfied by data for the data to be of interest to the subscriber.

6. The method of Claim 5, wherein the step of receiving the second query includes receiving the second query from a publisher in the subscription system, and

BNSDOCID: <WO_____03098479A2_I_>

wherein the second query specifies a second set of one or more criteria that must be satisfied by data for the data to be published to the subscriber.

7. The method of Claim 1, wherein the step of receiving the second query includes receiving the second query from a publisher in an information subscription system, and wherein the second query specifies a second set of one or more criteria that must be satisfied by data for the data to be published to a subscriber to the subscription system.

8. The method of Claim 1, wherein the step of representing a first conditional expression as data in a column includes representing a set of first conditional expressions from a plurality of first queries as data in the column, the method further comprising the steps of:
creating an index on the column of the table by
grouping predicates from the set of first conditional expressions based on predicate identifiers that are associated with respective criteria of the first set of criteria;
storing in a predicate table sets of predicate operators and constants in association with respective predicate identifiers;
creating a concatenated bitmap index based on a set of predicate operators and constants associated with a particular predicate identifier.

9. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the any one of the methods recited in Claims 1-8.

10. A method for evaluating an expression set that is stored as data in a column of a table, the method comprising the steps of:
classifying each predicate from each expression of the expression set as one of,
an indexed attribute predicate, wherein a bitmap index is created based on a set of operator and constant attributes that are stored in respective columns of a predicate table in association with a respective predicate identifier;
a stored attribute predicate, wherein a set of operator and constant attributes are stored in respective columns of the predicate table in association with a respective predicate identifier, and wherein no index is created on the set of operator and constant attributes;

BNSDOCID: <WO_____03098479A2_I_>

a sparse predicate, in which a predicate identifier associated with a sparse predicate is uncommon in the expression set, and wherein a sparse predicate is stored as data in the predicate table; and
filtering the expression set based on the classification of predicates from the expression set.

11. The method of Claim 10, wherein the step of filtering the expression set includes the steps of:
first, filtering the indexed attribute predicates to obtain a first set of expressions that includes one or more expressions for which all of its indexed attribute predicates are true for a given data item;
second, filtering the stored attribute predicates of the first set of expressions to obtain a second set of expressions that includes one or more expressions for which all of its stored attributes are true for the given data item; and
third, filtering the sparse predicates, if any, of the second set of expressions to obtain a third set of expressions that includes one or more expressions for which all of its predicates are true for the given data item.

12. The method of Claim 11, further comprising the step of:
publishing the given data item to a subscriber to an information subscription system that has expressed interest in data that meets criteria represented by an expression from the third set of expressions.

13. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the any one of the methods recited in Claims 10-12.

14. A computer-readable medium storing an index that is built on a table that has a plurality of rows, the index comprising:
concatenated bit vectors, wherein each bit vector is associated with one of an operator and a constant that are each stored in respective columns of the table; wherein the operator and the constant are associated with a predicate of one or more predicates from a conditional expression that specifies one or more criteria that must be satisfied by data for the data to be of interest to a subscriber to a subscription system.

BNSDOCID: <WO_____03098479A2_I_>

15. An apparatus for managing expressions in a database, comprising:
means for receiving a first query that includes a first conditional expression;
means for representing the first conditional expression as data in a column of a table;
means for receiving a second query that specifies a first set of one or more criteria; and
means for executing the second query to select data based at least on whether conditional expressions in the column satisfy the first set of criteria.

16. An apparatus for evaluating an expression set that is stored as data in a column of a table, comprising:
means for classifying each predicate from each expression of the expression set as one of,

an indexed attribute predicate, wherein a bitmap index is created based on a set of operator and constant attributes that are stored in respective columns of a predicate table in association with a respective predicate identifier;

a stored attribute predicate, wherein a set of operator and constant attributes are stored in respective columns of the predicate table in association with a respective predicate identifier, and wherein no index is created on the set of operator and constant attributes;

a sparse predicate, in which a predicate identifier associated with a sparse predicate is uncommon in the expression set, and wherein a sparse predicate is stored as data in the predicate table; and
means for filtering the expression set based on the classification of predicates from the expression set.

BNSDOCID: <WO_____03098479A2_I_>

NAME	AGE (102)	CITY (104)	INTEREST (106)
SCOTT	37	NEW YORK	SYMBOL ='CSCO' and PRICE > 25
ANDY	45	SEATTLE	SYMBOL ='MSFT' and CHANGE/PRICE > 0.1
STEVE	29	NEW YORK	SYMBOL ='ORCL' and PRICE > 30 and CHANGE > 5
ADAM	32	SAN JOSE	SYMBOL ='INTC' and CHANGE/PRICE < -0.1

FIG. 1

BNSDOCID: <WO_____03098479A2_I_>

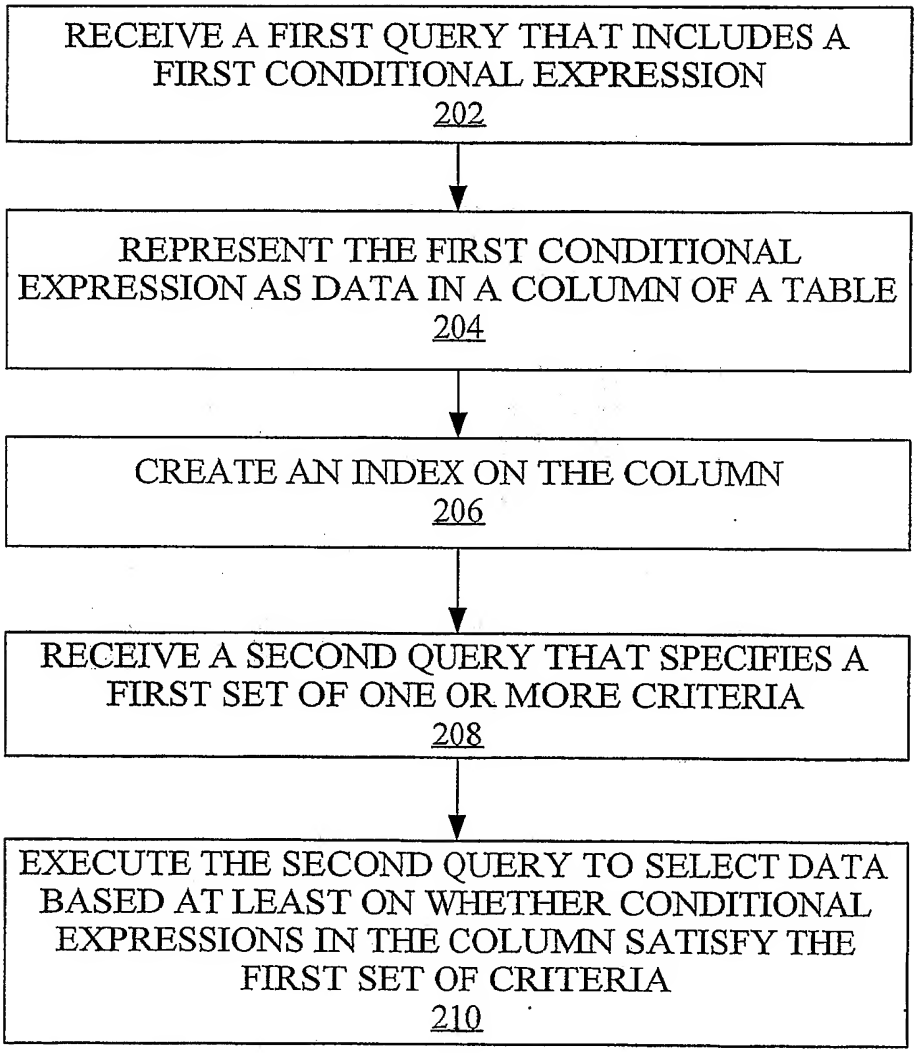


FIG. 2

BNSDOCID: <WO_____03098479A2_I_>

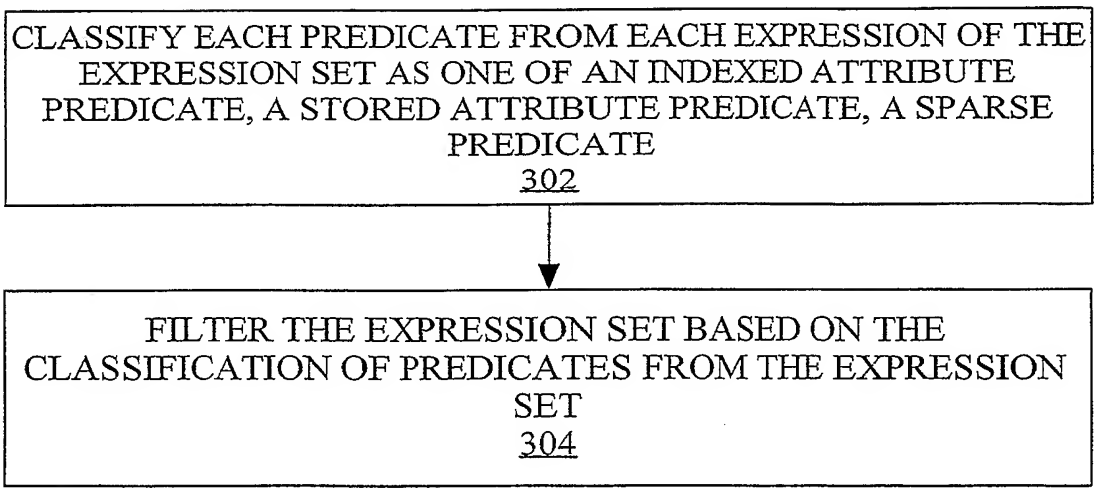


FIG.3A

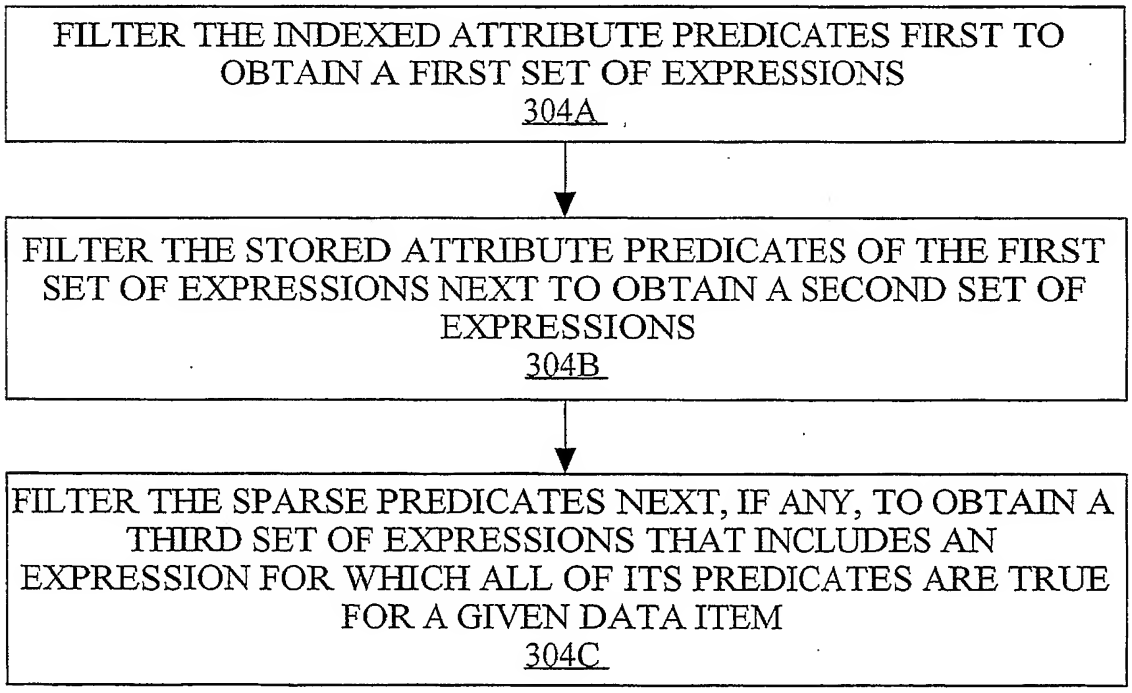


FIG.3B

BNSDOCID: <WO_____03098479A2_I_>

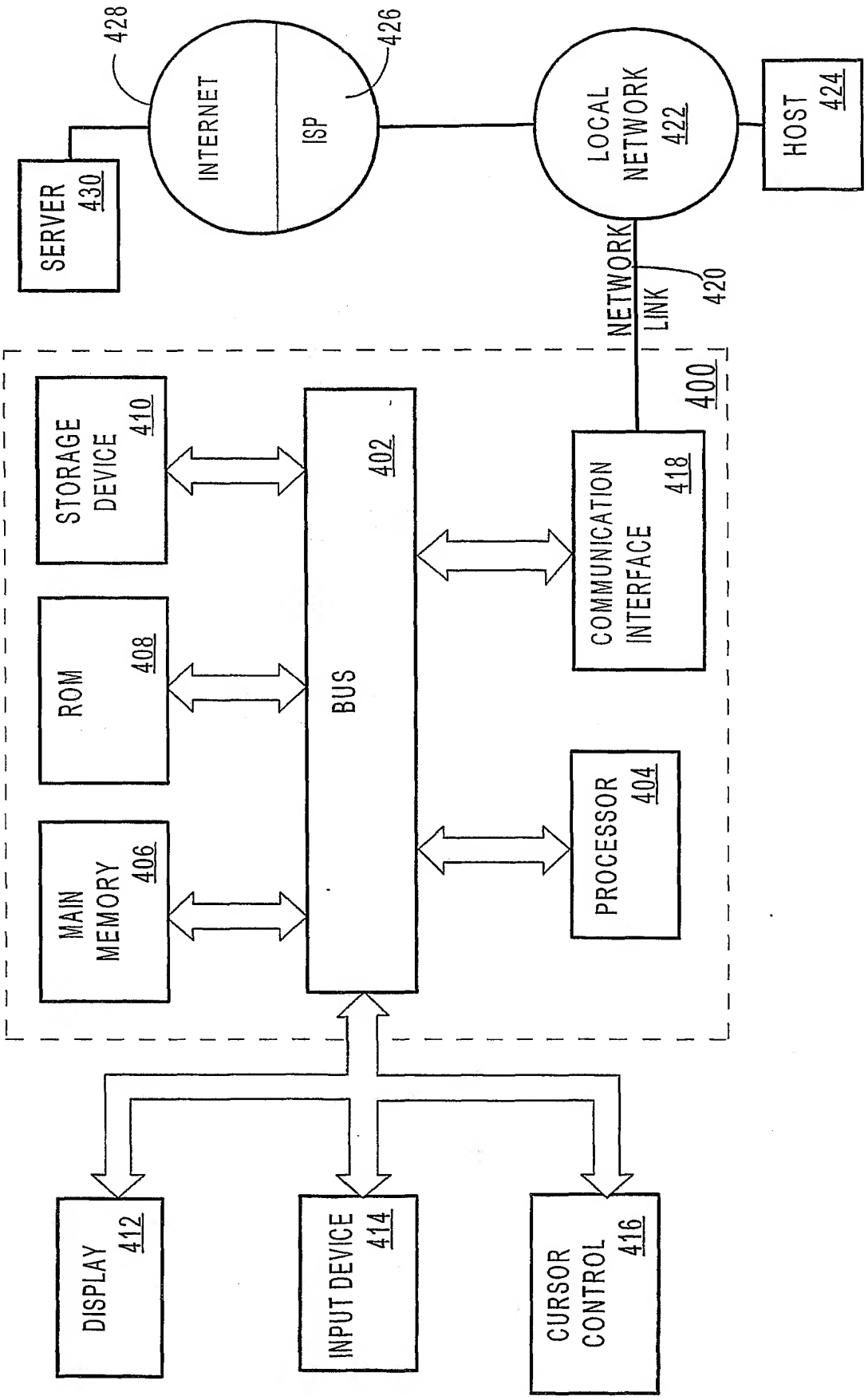


FIG. 4

BNSDOCID: <WO_____03098479A2_I_>

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
27 November 2003 (27.11.2003)

PCT

(10) International Publication Number
WO 2003/098479 A3

(51) International Patent Classification[7]: G06F 17/30

(21) International Application Number:
PCT/US2003/014892

(22) International Filing Date: 8 May 2003 (08.05.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/378,018 10 May 2002 (10.05.2002) US
10/254,383 24 September 2002 (24.09.2002) US

(71) Applicant: ORACLE INTERNATIONAL CORPORATION [US/US]; 500 Oracle Parkway, Redwood Shores, CA 94065 (US).

(72) Inventors: YALAMANCHI, Aravind; 8 Louisburg Square, Apt. 9, Nashua, NH 03060 (US). GAWLICK, Dieter; 757 Paul Avenue, Palo Alto, CA 94306 (US). SRINIVASAN, Jagannathan; 1 Hampshire Drive, Apt. F, Nashua, NH 03062 (US).

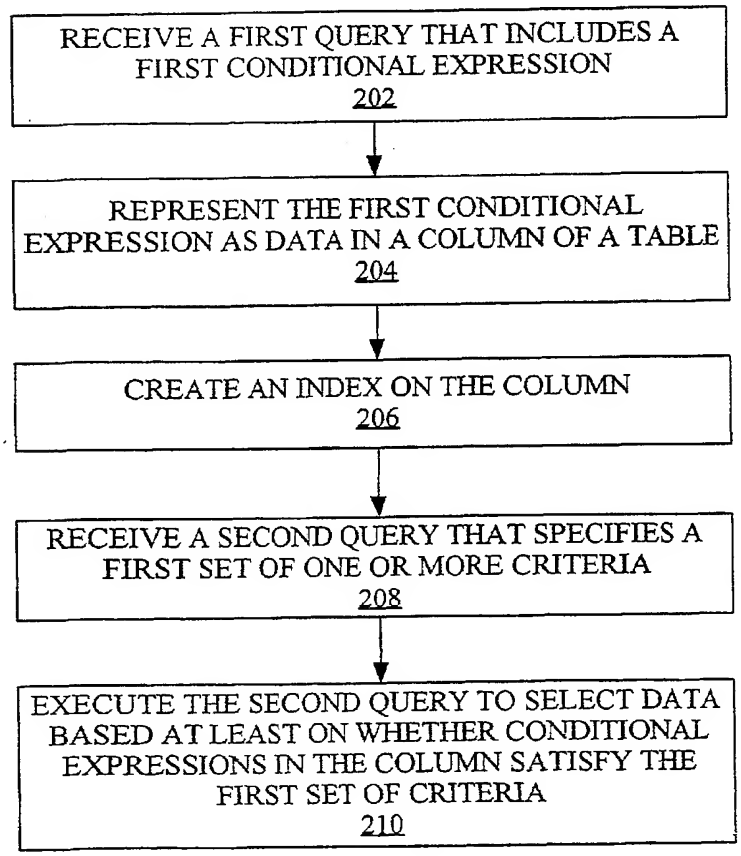
(74) Agents: HENKHAUS, John et al.; HICKMAN PALERMO TRUONG & BECKER LLP, 1600 Willow Street, San Jose, CA 95125 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE,

[Continued on next page]

(54) Title: MANAGING SEARCH EXPRESSIONS IN A DATABASE SYSTEM



(57) Abstract: Managing expressions includes receiving a first query that includes a conditional expression. The expression is then represented as data in a column of a table. A second query is received that specifies a first set of criteria, and the second query is executed to select data based at least on whether expressions in the column satisfy the first set of criteria. In an embodiment, the second query further specifies a second set of criteria, wherein executing the second query includes selecting data based on whether data in columns other than the expression column satisfy the second criteria. A special index is defined, which can be created on the column that stores the expressions, to filter large sets of expressions efficiently. A method of evaluating an expression set stored as data in a table classifies each predicate from each expression, and filters the expression set based on the predicate classification.

WO 2003/098479 A3

ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

(88) Date of publication of the international search report:
8 April 2004

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

BNSDOCID: <WO_____03098479A3_I_>

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 03/14892

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F17/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 959 416 A (IBM) 24 November 1999 (1999-11-24) claim 1 ---	1-7,9,15
X	GB 2 354 847 A (IBM) 4 April 2001 (2001-04-04) the whole document ---	1-7,9,15
X	GB 2 354 848 A (IBM) 4 April 2001 (2001-04-04) the whole document ---	1-7,9,15
X	WO 01 88795 A (ISHII KATSUYUKI ;TANAKA NENSUKE (JP); DAIWA SHOKEN GROUP HONSHA KK) 22 November 2001 (2001-11-22) abstract --- -/--	1-7,9,15

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
25 November 2003	12.02.2004

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	May, M

Form PCT/ISA/210 (second sheet) (July 1992)

BNSDOCID: <WO_____03098479A3_I_>

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 4 815 030 A (MOY DIANA Y ET AL) 21 March 1989 (1989-03-21) the whole document	1-7,9,15
A	PATENT ABSTRACTS OF JAPAN vol. 017, no. 435 (P-1590), 11 August 1993 (1993-08-11) & JP 05 089143 A (KOKUSAI ELECTRIC CO LTD), 9 April 1993 (1993-04-09) abstract	1-7,9,15
X	WO 01 08048 A (ORACLE CORP) 1 February 2001 (2001-02-01) the whole document	8,10-14, 16
X	US 5 072 405 A (THERIAULT DECEASED DANIEL ET AL) 10 December 1991 (1991-12-10) the whole document	8,10-14, 16
Y	US 6 067 540 A (HYDE JULIAN ET AL) 23 May 2000 (2000-05-23) summary of the invention	8,10-14, 16
Y	"IMPROVED RETE ALGORITHM - HASHING TECHNIQUES APPLIED TO PARTIAL MATCH MEMORIES" IBM TECHNICAL DISCLOSURE BULLETIN, IBM CORP. NEW YORK, US, vol. 35, no. 3, 1 August 1992 (1992-08-01), page 149 XP000326213 ISSN: 0018-8689 the whole document	8,10-14, 16
A	FORGY C L: "RETE: A FAST ALGORITHM FOR THE MANY PATTERN/MANY OBJECT PATTERN MATCH PROBLEM*" ARTIFICIAL INTELLIGENCE, ELSEVIER SCIENCE PUBLISHERS. AMSTERDAM, NL, vol. 20, 1983, pages 17-37, XP000374011 the whole document	8,10-14, 16

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 03/14892

Box I Observations where certain claims were found unsearchable (Continuation of item 1 of first sheet)

This International Search Report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the International Application that do not comply with the prescribed requirements to such an extent that no meaningful International Search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box II Observations where unity of invention is lacking (Continuation of item 2 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

-see additional sheet

1. ☒ As all required additional search fees were timely paid by the applicant, this International Search Report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this International Search Report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this International Search Report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest.

☒ No protest accompanied the payment of additional search fees.

BNSDOCID: <WO_____03098479A3_I_>

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. Claims: 1-7,9,15

   Event-based subscription system, where content publication requests are represented as database data and content subscription requests are represented as database queries. Problem: How to manage search expressions for an event-based subscription system.

2. Claims: 8,10-14,16

   Evaluation method using expression predicates and concatenated bit vectors for event-based subscription systems. Problem: How can expression predicates in a database table be efficiently indexed.

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/US 03/14892

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0959416 A	24-11-1999	EP 0959416 A2	24-11-1999
		CZ 9901481 A3	17-11-1999
		HU 9901539 A2	28-12-1999
		PL 332953 A1	22-11-1999
		US 6487548 B1	26-11-2002
GB 2354847 A	04-04-2001	NONE	
GB 2354848 A	04-04-2001	NONE	
WO 0188795 A	22-11-2001	AU 5677901 A	26-11-2001
		GB 2367407 A	03-04-2002
		WO 0188795 A1	22-11-2001
		US 2002152149 A1	17-10-2002
US 4815030 A	21-03-1989	AU 595354 B2	29-03-1990
		AU 7658487 A	14-04-1988
		CA 1292324 C	19-11-1991
		CA 1315891 C2	06-04-1993
		DE 3750941 D1	16-02-1995
		DE 3750941 T2	10-08-1995
		EP 0258867 A2	09-03-1988
		JP 3141943 B2	07-03-2001
		JP 63065536 A	24-03-1988
		JP 3247891 B2	21-01-2002
		JP 2001188742 A	10-07-2001
JP 05089143 A	09-04-1993	JP 2929042 B2	03-08-1999
WO 0108048 A	01-02-2001	US 6405191 B1	11-06-2002
		AT 247302 T	15-08-2003
		AU 5781000 A	13-02-2001
		CA 2379423 A1	01-02-2001
		DE 60004537 D1	18-09-2003
		EP 1210676 A1	05-06-2002
		JP 2003505792 T	12-02-2003
		WO 0108048 A1	01-02-2001
		US 6502093 B1	31-12-2002
US 5072405 A	10-12-1991	EP 0367377 A2	09-05-1990
		JP 2178736 A	11-07-1990
US 6067540 A	23-05-2000	US 6195656 B1	27-02-2001
		US 6266662 B1	24-07-2001

BNSDOCID: <WO_____03098479A3_I_>